Multiple Neural Network Model Interpolation

Daniel C. Chin and Albert C. Biondo
Johns Hopkins University Applied Physics Laboratory
11100 Johns Hopkins Road
Laurel Maryland 20723-6099
daniel.chin@jhuapl.edu and albert.biondo@jhuapl.edu
Phone : (240) 228-4956

Abstract

This paper presents an efficient method for extracting a multi-model interpolation function from a nonlinear system. The multi-model interpolation function consists of couple simplified time-varying models in neural-network structure to dynamically approximate the nature of the physical phenomena to be interpolated and extrapolated. The purpose of using the multi-model interpolation function is to perform a real-time approximation. This paper demonstrates the interpolation in a simulated environment, the underwater acoustic transmission loss generated from the NAVY-standard acoustic propagation-loss model ASTRAL, which is not suited to real-time operation. The interpolation includes initial learning period that is on the order of 20 minutes (more or less time depends on the size of the parameter intervals and the complexity of the ocean environment), and the subsequent interpolation speed will be measured in fractions of a second, a several orders-of-magnitude improvement over conventional calculations. In addition, for the example presented here, the interpolation error is within 1% of the actual transmission-loss value in a root-mean-square (RMS) sense.

Keywords: Multi-model Interpolation; Multi-objective SPSA; Nonlinear Interpolator; Neural Network; Nonlinear acoustic wave function.

1 Introduction

This paper develops a model-fitting technique to perform the interpolation and extrapolation of a nonlinear time-varying system. The development is demonstrated on the problem of transmission loss of underwater sound. The technique involves simplified time-varying multiple models, neural networks (NN), and multi-objective simultaneous perturbation stochastic approximation (MSPSA). The simplified models represent the local phenomena that change in time; NN projects the model variations; MSPSA trains the NN-weights. The MSPSA was first introduced in Chin [1] and is based on the simultaneous perturbation stochastic approximation (SPSA) developed by Spall [2]. A collection of applications of NN in adaptive control of nonlinear systems can be found in Ng [3]. The localized multi-model technique has shown accuracy and efficiency in the transmission loss interpolation.

The transmission loss function is highly oscillatory and quite variable. There is no simple representation available to describe the sound wave propagation accurately. The various local medium interactions and reflections give the function its erratic structure. An interpolation method suggested for time-variant systems in Gohberg (Ed.) [4, pp. 153-259] was too complicated and worked only on a single model. In a previous study, a linearized interpolation approach, a simple linear fit between observed data points, was suggested and tried in FY98 Progress Repot to DARPA [5]. Although the linearized interpolator would save computation time over the actual simulation calculation, the preparation of the base transmission loss curves and the massive amount of data handling ultimately make the linearized interpolation intractable. Also, the resulting interpolation errors were not uniform throughout the parameter space interval desired for the interpolation.

This particular model-fitting design uses two independent neural networks as the base of interpolation. The models are designed to fit the local physical phenomena and the NN's store the model variation information. The interpolator is expected to approximate the sound wave transmission loss accurately within the training area along the transmission pass, therefore the training process should provides NN the intermittent information. There are two ways to provide the intermittent information: 1) from an accurate model representation for the inverse estimation like the one introduced in Chin [1]; or 2) the intermittent observations derived from a base model like the ones discussed here for interpolation. The intermittent observations are accessible in most simulation packages; the utilization allows the model-fitting technique to use less number of simulated transmission passes and to gain more information in preparation of the interpolator.

In comparison with a simple linearized interpolator, the model-fitting technique described here requires longer time per interpolation, but takes orders of magnitude less preparation time. The interpolation time for the model-fitting technique in comparison with the detailed simulation time is negligible. The example presented in this paper shows that using 10 propagation loss curves is enough to train the interpolator for a large portion of the parameter space where interpolation is desired. The base-propagation loss curves for linearized interpolation would use order of magnitude amount more transmission curves to achieve a comparable level of accuracy. The ability to use a few propagation-loss curves to train the NN-weights for accurate interpolation makes the model-fitting technique desirable in planing a real-time simulation-training mission that was questionable for the linearized interpolator.

The NN-weight training procedure also is a very important task for the interpolator; it should consider the matches for intermittent points and the divergent of two different objectives in the two models. Given the variability and oscillatory behavior of the function, the training process also should have some checks and balances. One way to deal with the multiple-objective problem is summing these objectives and forcing them into a single objective algorithm. However, it is then very hard to find the balance among the objectives and the convergence speed, see Chin [1]. The MSPSA algorithm introduced in Chin [1] optimizes the independent model parameter sets (the parameters in one set have no relationship with the parameters in other set) from relevant objectives and is suitable for training the NN-weights here. Also, the parameter dependencies are different for the two models, the algorithm could tailor the minimization procedure to accommodate the differences. In the end, MSPSA used small number of detailed simulation curves to train the interpolation functions, achieved acceptable root-mean-square errors from the original simulation results, and made real-time operation feasible.

2 Underwater Sound Transmission Loss

Follow the principle of underwater sound in Urick [6], the sound propagating through the ocean was described in three physical phenomena:

- Sound spreads while it propagates through the medium in three different ways: spherically, cylindrically, and linearly.

- The medium absorbs sound energy, with the rate of absorption varying with the water temperature and the acoustic frequency.

- Sound signals are also influenced by the reflections from the top and bottom of the ocean water column. This influence is a function of the local bottom bathymetry and composition, as well as the sea surface conditions (wave height).

These three effects also vary with the frequency of the sound signal.

Using two models could describe these three phenomena. One model approximates the energy spreading and absorption because their equations are similar; the other model approximates the reflections. The energy dissipation models dependents on water temperature at the referenced local areas; the reflection model depends on both site structures and range from the sound source.

The transmission losses are represented as a ratio of the sound intensity at a given range, say p nmi, to the intensity of a reference range. If TL represents the transmission loss and T represents the signal intensity, the sub-indexes represent the values evaluated at the reference points, respectively. Then, the relation between the transmission loss and signal intensity are as following:

$$TL_p = 10\log\left(\frac{T_p}{T_0}\right). \tag{1}$$

The intensity T changes along the transmission pass in a non-linear pattern. For easier to approximate the intensity reduction, the sound propagation (the thick line inside the box on Figure 1) is divided into a fine fixed-interval grid, the grid points are located at unit marks for convenience of data handling, in our study here uses nautical mile marks. The marks 0, p on Fig. 1 are the reference points near source and at receiver. The marks $1, 2, \cdots, k$ are the reference points on the grids.

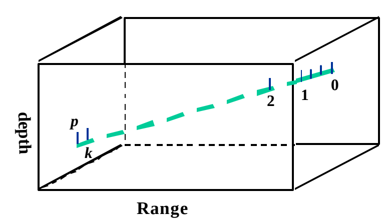

Figure 1: The reference points and sound propagation pass

The total sound transmission loss between source and receiver could be expressed as the accumulation of transmission losses along the pass, as shown in (2); an expansion of Equation (1):

$$TL_p = 10\log\left(\frac{T_p}{T_k}\right) + \left[10\log\left(\frac{T_k}{T_{k-1}}\right) + \cdots + 10\log\left(\frac{T_1}{T_0}\right)\right]. \tag{2}$$

The examples in Section 5 use the nautical mile as grids and 3-ft as the initial 0 reference points. The receiver mark p is located between grids k and k+1 include k+1.

At each reference points, $i \in \{0,1,\cdots,k,p\}$, the transmission loss is also the total effect from all three individual physical effects that are represented by two models. Let T_i^E and T_i^R be the signal intensity reduction value due to energy dissipation and reflection between the reference points $i-1$ and i. The total sound wave intensity at the reference point i would be

$$T_i = d\left(T_i^E - T_i^R\right). \tag{3}$$

where d is the distance between interpolating position and reference point $i-1$, in (3) the interpolating position is the position of reference point i. Substitute $T_i \, \forall i$ into (2), and then computes the total transmission loss from source to receiver.

To simulate T_i from detailed non-linear models are both computational and computer I/O intensive operations. Simplification of the data structure and retrieval system are the first couple steps in reducing the computational burden and data handling problems. Because mass amount of data for the water temperature profiles and detailed ocean basin information along the transmission pass are required in computing the transmission losses from the non-linear accurate models. This paper utilizes a pair of NN for so purpose.

The equation for energy dissipation intensity reduction formula T_i^E is a simple constant varying mostly with the water temperature (assuming uniform value within two reference points) and could theoretically have value range between 0 and 3 (not including 0). For generality, the equation is expressed in two degree of freedom among T_i^E 's as following:

$$T_i^E = s\left(1 - \frac{r}{2}\right), \tag{4}$$

where s and r are two constants with values between 0 and 1 that will be the output from NN (the outcome could be in expected range for the interpolation area of interests, instead of the theoretical range). There is no simple expression for T_i^R. This paper uses the first order of trigonometric function to represent the energy gain from reflection as following:

$$T_i^R = a\cos\left(\frac{p\,fp}{720}\right) + b\sin\left(\frac{p\,jp}{720}\right), \tag{5}$$

where a, b, f, and j are four coefficients and would be the output from NN, π is the radian constant, $3.14159\cdots$. This equation may be changed due to environmental differences, e.g., with a higher order representation for a more complicated environment.

The neural-networks are designed for tracking the variations of the coefficients used in (4) and (5) among the intensity reduction functions due to environmental change along the transmission pass. The neural-network for the energy dissipation model is a two-hidden layer network with four inputs and two outputs. The number of weights for each of the two-hidden layers is five. The four inputs are frequency, source depth, depth at the initial reference point, and delta range from the interpolation point to the initial reference point. The two outputs denoted by s and r are the spreading factor and absorption rate as defined before. The neural-network for the reflection model is a one-hidden layer network with four inputs and four outputs. The number of weights for the hidden layer should be changed according to the size of the geographical area, the bottom type and the sound frequency; a larger area, more complex bottom types and higher frequencies will use a larger number of the NN-weights. The inputs for this network are frequency, source depth, depth at the initial reference point, and range of the range from source. The outputs for this network are the coefficients of a trigonometric equation.

3 Interpolation Setting

The underwater sound transmission loss can be expressed in a more general term, such as a system y consists of two models denoted by $f(\bullet)$ and $g(\bullet)$, and

$$y = F\left(f(\bullet), g(\bullet)\right), \tag{6}$$

where the function F is nonlinear and model f and g are varying with time. The value of y can be accumulated from a sequence of intermediate function values y_i and

$$y = \sum_{i=0}^{p} y_i \tag{7}$$

where y_i is evaluated at the reference point i and $i \in \{0, 1, \cdots, k, p\}$. Reference points 0 and p are located at the boundary points; reference points $\{1, \cdots, k\}$ are located at the internal grid points along the transmission pass. The individual y_i is also a function of $f_i(\bullet)$ and $g_i(\bullet)$ and

$$y_i = F\left(f_i(\bullet), g_i(\bullet)\right) \tag{8}$$

Assuming the functions $f_i(\bullet)\,\forall i$ can be approximated by the same function with different coefficients such as the one in (4), likewise for $g_i(\bullet)\,\forall i$ as the one in (5).

Let NN denote a neural network, with NN_f and NN_g the neural networks for models f and g, respectively. Assume $x_{f,i}$ and $x_{g,i}$ are the input terms of NN_f and NN_g at the interval between reference points i-1 and i. Then,

$$\begin{aligned} x_{f,i} &\to NN_f \to f_i \\ x_{g,i} &\to NN_g \to g_i \end{aligned}, \tag{9}$$

where f_i and g_i are the neural network output parameters associated with the two models at the same interval and will be used as the coefficients of $f_i(\bullet)$ and $g_i(\bullet)$. Then these two functions could be defined as $f_i(x_{f,i} \mid w_f)$ and $g_i(x_{g,i} \mid w_g)$, where w_f, w_g are the weights of NN_f, NN_g. Function $f_i(\bullet)$ is the function of $x_{f,i}$ based on the weight values of NN_f; similarly function $g_i(\bullet)$ is the function of $x_{g,i}$ based on the weight values of NN_g. Let $\hat{q}_f, \hat{q}_g$ be the estimated variables for w_f, w_g and $\hat{y}$ be the approximation value for y and

$$\hat{y} = \sum_{i=0}^{p} F\left[f_i\left(x_{f,i} \mid \hat{q}_f\right), g_i\left(x_{g,i} \mid \hat{q}_g\right)\right]. \tag{10}$$

We are trying to minimize $(y-\hat{y})^2$ and combinations of $(y_i - \hat{y}_i)^2$ for all transmission loss curves to find the best fit $\hat{q}_f, \hat{q}_g$ of w_f, w_g. Then we could use $\hat{y}$ as an interpolation value from the given sets of input $\{x_{f,i}\}$ and $\{x_{g,i}\}$, i.e. the input parameters defined at the reference points along sound transmission pass as they are define in Section 2. This setting may easily be expanded into a system that involves more than two models and may also be used in a control environment.

4 The Training Algorithm

The multiple-objective simultaneous perturbation stochastic approximation (MSPSA) algorithm presented in [1] is used to train the neural-network weights. The algorithm attempts to minimize the sum of the square difference between interpolation values and the computed values over the local intervals, the reference points assumed in the derivation of the equations, as well as the entire data range. The differences are calculated at each computed value, according to the resolution of the data. The minimizations are completed over iterations. Any single iteration consists of many small steps from the individual minimizations that are sequenced one-by-one; let us call the small step a minimization step. The estimates of one minimization-step will be passed to the next step in the sequence as the previous estimates of that step. The estimates from the last step in the sequence will be the estimates of the iteration. The estimates of the iteration will be passed to the first minimization step in the next iteration as the previous estimate of that step. This optimization algorithm assumes the data was generated with a consistency setting, similar environment or limited interference. Simulation data has less of a consistency problem then the real data. Even the inconsistency does exist among the data, the order of the step sequence will not affect the outcome of the estimates, and it just effects the convergence speed. The minimization-step procedure uses the equation stated in [2] as an iteration of the SPSA algorithm, using two statistical perturbation estimates to approximate a gradient that updates the previous estimates.

For convenience, let f represent the energy dissipation model and g represent the reflection model for the underwater sound transmission loss system. The subscript-index i indicate the local ranges along the transmission loss curve. Figure 2 shows the detailed training procedure of "one iteration" as follows:

1) Starting from the initial estimates or the previous iteration estimates, the first step (box 1) is to minimize the local differences between y_i and $\hat{y}_i\left(f_i\left(x_{f,i} \mid \hat{q}_f\right) \mid g\left(x_{g,i} \mid \hat{q}_g\right)\right)$ on the weights of NN_f while holding the weights of NN_g unchanged.
2) Using the 1st step estimates as the initials, the 2nd step (box 2) minimizes the full range difference between y and $\hat{y}\left(g_i\left(x_{g,i} \mid \hat{q}_g\right) \mid f\left(x_{f,i} \mid \hat{q}_f\right), \forall i\right)$ on the weights of NN_g while holding the weights of NN_f unchanged.

3) Initializing from the 2nd step estimates, the 3rd step (box 3) minimizes the summation of the square differences between y_i and $\hat{y}_i\left(f\left(x_{f,i}\,|\,\hat{q}_f\right), g\left(x_{g,i}\,|\,\hat{q}_g\right)\right)$ on the weights of NN_f and NN_g for a global fit.

4) Box 4 – indicates repeating steps 1, 2, and 3 for all source-receiver (SR) pairs.

5) Box 5 evaluations the estimates of weights on both NN_f and NN_g. If the estimation error, e , is greater than $1.20e_{\min}$, the last known smallest error, then reject the estimates and repeat the previous 4 steps. If the estimation error is within $1.20e_{\min}$, then update the weights on both NN_f and NN_g and proceed to the next iteration.

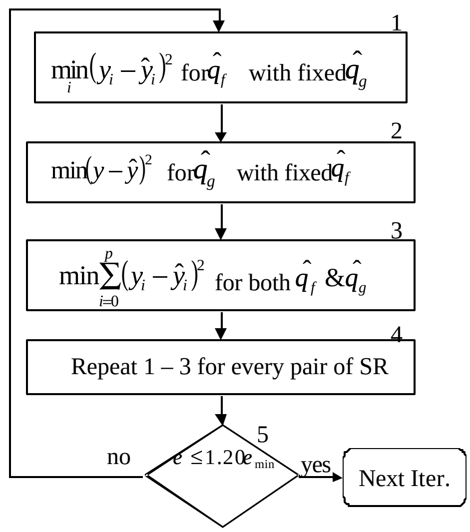

Figure 2 One Iteration of Training Algorithm

5. Example

ASTRAL (the Automated Signal Excess Prediction System (**AS**EPS) **TRA**nsmission **L**oss), is a Navy standard model, included in the Navy's Ocean and Atmospheric Master Library (OAML). OAML is a collection of configuration-controlled models and databases, maintained by the Naval Oceanographic Office (NAVOCEANO). ASTRAL was specifically designed to run rapidly, and is commonly used in real-time simulations because it runs 10 to 1000 times faster than the traditionally more accurate parabolic equation (PE) models, as well as other research models.

ASTRAL is primarily a range-dependent, adiabatic, range-smoothed mode theory model, with additional separate algorithms to model important acoustic features that are not appropriately handled in the primary algorithm. In particular, ASTRAL uses separate algorithms for convergence zone and surface duct propagation [6]. ASTRAL can predict the range averaged transmission loss and vertical angular arrival structure, but only the former quantity is considered here.

The selected model is expected to be run for all propagation calculations required during the Navy simulations in which is used. Therefore, ASTRAL was run for a variety of environmental conditions and operational parameters deemed reasonable for the simulation.

Assuming a simulation for purposes of real-time operator training, the oceanographic environment for this example was located in the Sea of Japan. For simplification, a single set of propagation paths with varying bathymetric details was used in the example. The differences between each of the paths were in the receiver depth, source depth, sound frequency, and transmission range. We assumed that a receiver was placed at certain discrete depths in a 200-ft interval from the surface, with the source placed at various depths between the surface and 1500-ft. Each source and receiver pair was generated using different frequencies, from 20 Hz to 10000 Hz. The total transmission range was 102 nmi; the resolution to which ASTRAL generated was 0.25 nmi.

The receiver depth, source depth, frequency, and transmission range define the parameter space interval desired for the interpolation, or the interpolation area, which also defines the real-time operator training area. In order to have the interpolator work properly; it has to learn the characteristics of the transmission loss from the data computed within the interpolation area. The main criterion for data selection is that the selected data has to contain all the important features in the area. For the time being, we were using trial and error to select 10 source-receiver pairs, each of them at a different frequency. Figure 3 shows the transmission loss surface formed by the 10 selected source-receiver pairs. The Y-axis in the Figure indicates the 10 selected pairs from 1–10; the X-axis indicates the total transmission range from 1 – 102 (102 nmi in 0.25 nmi resolution); and the Z-axis shows the scales of the transmission losses from 50 dB to 100 dB.

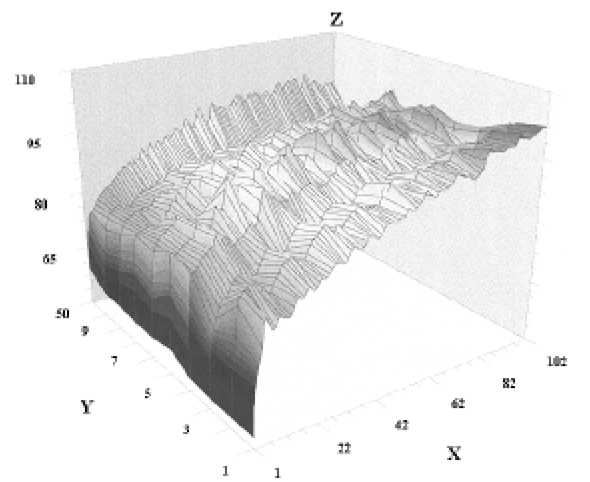

Figure 3 ASTRAL Generated Transmission Loss Surface

Figure 4 shows the transmission loss surface formed by the interpolation values at the same grid points as in Figure 2. The interpolator learned the features from the first half of the range data, within 50 nmi. Figures 3 and 4 show the matching surface on the left half of the surfaces (shorter ranges < 50 nmi), while missing some characteristics on the right half of the surfaces (beyond 50 nmi). When we included all of the range data to train the interpolator, the spike on the top-left end of the surface in Figure 4 was clearly shown on the surface formed by the interpolation values resulting from that interpolator. The RMS error for the transmission loss surface in Figure 4 is 0.5 dB, about 1 to 2% of the actual loss values. The interpolator surface showed in Figure 4 takes 360 iterations, for real-time operation 200 iterations would be sufficient for a 25-nmi range operation area; the RMS error for the smaller range operation area was about 1 dB in the case mentioned in the abstract. Using the more detailed simplification models, $\hat{f}$ and $\hat{g}$, would have less RMS errors, but they will take a longer time to train.

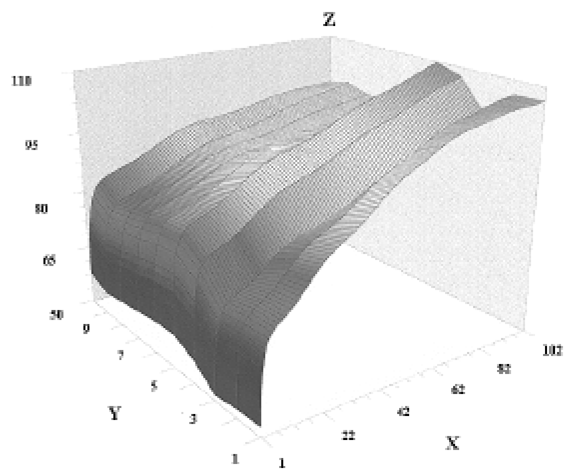

Figure 4 Interpolated and Extrapolated Transmission Loss Surface.

6. Summary

This paper presents a model-fitting technique for use as an interpolator for underwater acoustic transmission loss. This interpolator could be useful in creating function approximation for one local domain in the adaptive interpolator discussed in Spall [8]. The error of this interpolator was tolerable and the computation speed was adequate for real-time training operations. There is a tradeoff between accuracy and desired speed — the more accuracy required the more time required for the NN to train. There is also a tradeoff between the accuracy and the complexity of the operational area. There is room for improvement in the optimum source-receiver pair selection and in the estimation model formulations. This technique could be used for extrapolation and inversion processes.

Acknowledgements

This work was supported by DARPA under contract MDA972-96-D-0002 in support of the Advanced Simulation Technology Thrust Area, Multiresolution Interaction Validity Project.

References

[1] Chin, D. C., "Simultaneous Perturbation Method for Processing Magnetospheric Images," 1999, *Optical Engineering*, v. 4, pp. 606-611.

[2] Spall, J. C., "Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation," 1992, *IEEE Trans. Autom. Control*, v. 37, pp. 332-341.

[3] Ng, G. W., Editor, *Application of Neural Networks to Adaptive Control of Nonlinear Systems*, 1997, Research Studies Press. LTD. & John Wiley & Son Inc., ISBN 0471 97263 0.

[4] Gohberg, I., Editor, *Time-Variant Systems and Interpolation*, 1992, Birkchasuer Verleg Basel, ISBN 0-8176-2738-3.

[5] "FY98 Progress Report; Multiresolution Interaction Validity Project," 1998, *Advanced Simulation Technology Trust*, STD-R-2774.

[6] *Software Test Description for the ASTRAL Model*, Naval Oceanographic Office, NRL OAML-STD-23E, 1994.

[7] Urick, R. J., *Principles of Underwater Sound*, 1976, McGraw-Hill, ISBN 0-932146-62-7.

[8] Spall, J. C., "Adaptive Model Fitting with Time-Varying Input Variables," 1999, *The Proceedings of 1999 American Control Conference*, San Diego, CA, pp. 1435 - 1440.